Cranfield UNIVERSITY

Applications in Practical High-End Computing - Group Project

Assignment - "Workflow"

Project Management

Supervisor: Dr Stuart Barnes

Authors: Mateusz Gołąb
Csaba Kerti
Jakub Kiełbasa
Zsolt Kollarits

Course: CSTE / SETC

Table of contents

1. Introduction

During our works on Workflow Recovery project we have been obligated for creating documents representing different aspects of our project. Documents have been prepared by different team members, which sometimes make cause ambiguities.

On the other hand, during our work we have tried to applied quite modern project management methodology which is was worth to be described.

Mentioned above reasons, plus fact that we would like to summarise our work some way, convinced us that it is necessary to create additional document describing how our team - Eastern Eagles - works.

Another reason, explaining significance of this document is our trial to estimate costs of our work and trial to forecasting remaining amount of work.

a) Eastern Eagles Team

Our team is build on base of handful of people who have some experiences in Software Engineering and Software Development. We know each other from the first classes, and, since we use to think that our technical skills are quite similar we decided to start cooperation. We found it quite reasonably also because we represent two different countries and this working on this project can improve our communication skill.

We are extremely sad about the fact, that because of time pressure we did not manage to prepare official logo and introduce it in this document.

2. Literature review

Projects responsible for controlling and managing workflows are not very common. They are mostly designed for small groups of people, usually scientists. This fact causes situation where accessing proper literature is not so trivial.

In all our stages we have been using general software engineering sources, and we have tried to find books connected specifically to workflows' topic - in this matter, we have found, and reused mostly Design patterns. All used source are mentioned in specific documents in References' or Bibliography's chapters.

We do not claim that our review is, even partially, completed. Unfortunately, performing wider one would take weeks, and since we had less than 3 of them for entire work, we found it not, that ,critical.

3. Scrum Methodology

a) Theory

Scrum itself is an example of Agile methodology. It is quite new approach to project management issue. It fulfils all Agile manifest's assumptions. Scrum should be used in case of small and medium size projects because in the largest ones whole idea is being misused because of the size of a team.

Few facts about Scrum:

- Whole life-cycle of project is divided into *Releases*.
- *Releases* are divided into iterations, in this methodology called *Sprints*.
- *Sprint* should usually be not longer than 30 days and not shorter than 5 days.
- *Release* should contain between 4-12 *Sprints*.
- List of tasks in *Release* is called *Product Backlog*
- List of tasks in *Sprint* is called *Sprint Backlog*

Roles in Scrum:

- Scrum Master - person who gives tasks to others and organize work
- Development Team - all persons responsible for performing tasks
- Product Owner - usually external customer

Events:

- Daily Scrums - daily meetings (usually standing up) where each member of Scrum Team answer 3 questions:
  - What have I done since last Scrum?
  - What impediments have I met?
  - What am I going to do next?
- Sprint review/planning meetings - at the beginning and at the end of each Sprint Scrum Team is reviewing results of last works and planning next Sprint Backlog.

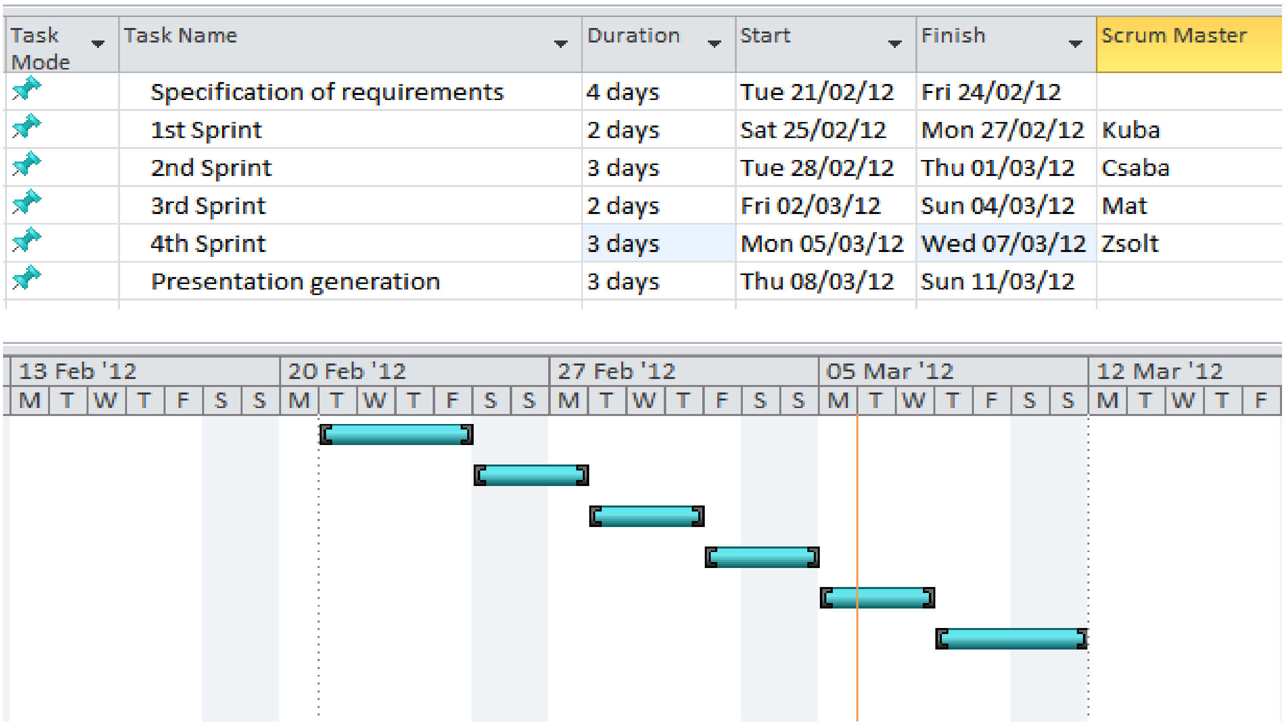
b) Why Scrum?

Main motivation for starting using Agile techniques was quite simple - it is very popular in contemporary, developing world and most of companies working with Software Developing are using it. Since none of us was fluent in this matter we decided that it is worth a shot. Other reason was, that Scrum is actually simple and well-defined way of managing small team, and because we were very short in time, we have realized that we do not have weeks for learning more sophisticated methodologies.

c) Releases, Sprints

Because of the duration time of the project there was actually no point of making any division into several *Releases*.

Release has been divided into 4, very short Sprints. The reason for such a situation was that there are 4 students in our team, and 4 main responsibilities (requirements, design, implementation, tests), this way we hoped that we could focus on one job at the time. Another reason for that choice is placed in another header - about Roles.



Task Mode	Task Name	Duration	Start	Finish	Scrum Master
📌	Specification of requirements	4 days	Tue 21/02/12	Fri 24/02/12	
📌	1st Sprint	2 days	Sat 25/02/12	Mon 27/02/12	Kuba
📌	2nd Sprint	3 days	Tue 28/02/12	Thu 01/03/12	Csaba
📌	3rd Sprint	2 days	Fri 02/03/12	Sun 04/03/12	Mat
📌	4th Sprint	3 days	Mon 05/03/12	Wed 07/03/12	Zsolt
📌	Presentation generation	3 days	Thu 08/03/12	Sun 11/03/12	

13 Feb '12
20 Feb '12
27 Feb '12
05 Mar '12
12 Mar '12
M T W T F S S M T W T F S S M T W T F S S M T W T F S S M T W T F

Gantt chart generated in MS Project.

d) Roles

Development Team - main problem with dividing roles between us four was that (because of time) we have to use all of our human resources for hard work in Development Team. (Which was actually not only developing but also designing, testing etc.)

Scrum Master - natural problem was lack of Scrum Master when all workers are being used as a developers. We have solved this problem by moving position of Scrum Master within us. (That is another reason why we have decided to split *Release* into 4 *Sprints*.) Each of us was holding the Scrum Master position during one Sprint.

Product Owner - we have mocked our customer by sending them mails with questions which appeared during our work.

e) Meetings

Daily Scrums was very convenient way of exchanging information between team members. Ideally, Scrum should last about 15 minutes in a bigger team than ours. We have not achieve this number (record was 4 and half hour), but, as time went we get better with organisational issues. Another problem was the time of meetings. Ideally it should be always the same hour, but of course, it was impossible because of other classes etc. Because of time pressure we have decided to meet each day including Saturdays and Sundays.

f) Scrum Summary

We have tried to apply Scrum methodology to our project and outcome was generally not bad. During these few weeks we get familiar with general rule of whole process and get some experiences about Agile techniques. Unfortunately, we have met many impediments which have complicated our trials in this matter.

4. Responsibilities/Roles of team members.

During work on this project we have to split responsibilities for different aspects of it to different team members. Although, we have assigned each stage of project to separate persons, we all have been working on each of the stages. This mechanism was quite simple - one persons responsible for particular stage was controlling and monitoring it, explaining what else should be done, and all others were helping him in fulfilling this duties. We have, indirectly, connected this with our Scrum meetings (we have tried to achieve situation where Scrum Master would be a person responsible for, currently, most important stage).

One of main responsibilities of a person responsible for given stage was to prepare reliable report. Another, even more important, aspect was that this person should pass very clear information to person responsible for following stage about situation of the project (very careful explanation about all details connected to it).

Below, we will try to briefly describe more precise responsibilities of particular people.

a) Project Manager

Each of us four was Project Managers for some time. We achieved this situation by mixing Scrum methodology and responsibilities division. When we started a stage for which one of us was responsible, he become Project Manager and was making most important decision. This approach was quite risky, because in mean environment one person can overuses their power as Project Manager e.g. because somebody else was giving them too many tasks.

We have avoided this situation by spending a lot of time on communicating among team. There was (hopefully) no single situation when person responsible for some task was (explicitly) unsatisfied because of it. Price for this situation was quite high - we have spent usually not less than 2 hours on daily discussing about project.

Very good thing about this approach was that all of us has chance to meet problems and concerns connected to managing human resources - which was even better solution because of the fact that none of us was assigned as a Project Manager by lecturer - taking this position by any of us would be simply unfair.

b) Requirement Manager - Jakub Kiełbasa

First stage of each project is about collecting requirements, doing feasibility study and form requirements into more formal shape, so that another stages' Managers would be clear about them. Jakub was responsible for that part. He has been controlling process of forming Functional and Non-functional requirements as well as controlling process of creation and designing Use Case Diagrams. Most of this jobs was very strongly affecting Design Stage and Test Stage as well. He was probably most hated person within team.

c) Test Manager - Csaba Kerti

Stage of Testing was probably most complicated one because, contrary to all others, there is no way of chronological placing it strictly between two other stages. It was continuous process which started just after Requirements' part finished and finished just after Implemented part finished. Person responsible for this stage was Csaba. He was responsible for creating acceptance tests - connected to requirements, integration tests - design and unit tests - implementation. He was probably most frustrated person in the team.

d) Design Manager - Mateusz Gołąb

Stage of Design follows the stage of Requirements and precedes stage of Implementation. It was about creating technical, design of system being created. It includes modelling whole workflow system and find out most appropriate design pattern for our case. Person responsible for this stage was Mateusz. He was appropriate person for this function since he was famous because he organises even his clothes into units. He was controlling process of creation of logical diagrams of project and was responsible for system configuration. He has to assured that project Design will fulfil all Requirements gathered in previous stage. This job strongly affects Implementation Stage.

e) Implementation Manager - Zsolt Kollarits

Stage of Implementation may be considered as finalization of our work. If our System was prepared correctly and efficiently than Implementation should be fairly easy otherwise this stage may be a torment. Person responsible for this part was Zsolt, who strongly refuses to accept the fact that previous stages was prepared excellently. Because of the fact that the entire project Implementation would take millennia we had to choose part of it to implement as a Prototype. He was in charge of it. Implementation form was strongly impacting part of Test i.e. unit tests.

5. Costs estimation

Estimating costs connected to a software project is probably one of the firsts and one of most important activity during feasibility studies. It is essential to deliver customer concrete figures stating how much will project cost - in both: time and money terms. Below we have tried, using strange techniques and common senses, estimate this number.

a) Costs of finished work

During preparations of this prototype of project we have spent concrete number of hours doing different activities. Below we have placed some explanations and final number of the cost of current version of our product.

Activity\Dimension	Hours:	Hour's cost:	Total cost:
Requirement Analysis	72	£60.00	£4,320.00
Software Design	64	£60.00	£3,840.00
Software Development	51	£40.00	£2,040.00
Quality Assurance	51	£30.00	£1,530.00
Project Management	34	£80.00	£2,720.00
Total sum:	272		£14,450.00

Comment 1: We have been working from 21 Feb till 8 March i.e. 17 days.

Comment 2: Each of us was working approximately 4 hour a day.

Comment 3: Hour's cost is our estimation of the market value of a working hour based on job offers.

Comment 4: Hour's cost is final number - it includes such a things as maintenance of the bureau etc.

b) Costs forecasting - "how much to finish"

Another part of cost estimation is much harder. We will try to forecast how many working hours do we need for finishing our product. Then, based on that number we will try to predict was would be final cost of entire project.

Activity\Dimension	Hours:	Hour's cost:	Total cost:
Requirement Analysis	160	£60.00	£9,600.00
Software Design	480	£60.00	£28,800.00
Software Development	1920	£40.00	£76,800.00
Quality Assurance	800	£30.00	£24,000.00
Project Management	640	£80.00	£51,200.00
Total sum:	4000		£190,400.00

Comment 1: As we can notice project would take 2.5 year for single multitalented person.

Comment 2: In current shape of our team we cannot finish this project faster than in 7 months.

Comment 3: It is possible to accelerate this process by employing new workers, but since it is risky venture it will cost additional £10,000.00 per each new employee, each new employee would accelerate work by 1 month.

Comment 4: Dependencies between stages exist, so that we find impossible to finish project faster than in 4 months despite number of new employees - so that there is no point in increasing team size above 7 people.

Comment 5: Hour's cost is still final number! It includes such a things as maintenance of the bureau etc.

c) And the magic number is:

Total cost of the project is:

£204,850.00

If customer would decide to accept offer immediately and would pay by cash we can give them discount, so that final cost would decrease to just:

£200,000.00

This is one of this moments where words are simply not enough.

6. Notations

In order to simplify communication within our team we have decided to introduce some notations, which should decrease number of misunderstandings. Another point of this idea was to simplify documentation process - without unique identifiers it would be very ambiguous. Below we describe few notations used within our project.

a) Use case

UC_XY - typical use case identifier. Where:

X - stands for actor most connected to current use case.

Acronyms are: U - User, A - Administrator, S - Scientist, M - Managers.

Y - index of use case for X actor.

Examples:

UC_A3, UC_M1.

b) Requirement

FUX - functional requirement where X is requirement index.

NUX - non-functional requirement where X is requirement index.

Examples:

NU1, FU4.

c) Test case

TC_XXXX - typical test case identifier. Where:

XXXX - stands for unique number of test case

Examples:

TC_1100, TC_1300.

d) Dictionary

In our project we have defined Dictionary of terms in order to avoid as many misunderstandings as it is possible. It is included in Requirements document. It is being used in all other documents. For distinguishing its terms from regular words we start them from block letter.

Examples:

Workflow Manager, Scientist.

7. Used Tools

a) Enterprise Architect ver. 9.2 Trial Ultimate

Main tool, used in creating System Requirements and in further stages was mentioned above program. It is a very advanced, powerful tool which allows software engineer creating all kinds of UML diagrams and assures support and necessary documentation. We agreed that, for so important project we will use Ultimate edition of it - there was no place for risk.

This software is quite expensive (Ultimate edition costs $849) but vendor delivers free Trial 30-day version which was perfect for us, since project's life-cycle was about 20 days.

All diagrams presented in this document are prepared within EA. After few weeks of usage we can recommend it.

b) Tortoise SVN ver. 1.75

Since we have decided to use Version Control in our project we have to choose server where our code would be stored and client for accessing it. As a client we decided to use mentioned above program.

It is free, user-friendly software which gives quite a lot of possibilities. It is very easy to locate source change, compare different files etc. It also supports all authentication mechanisms.

Only problem with this tool was cooperating with Enterprise Architect - unfortunately project prepared in EA is represented as a single physical file. It caused some problems with merging our works.

c) Google Code

As I server where we could store our whole repository we have chosen Google Code. It is free, very simple in usage program which is especially convenient for persons who use Google accounts for other purpose - whole authentication is then automatic. It can be easily used together with Tortoise SVN so that there are no problems with incompatibilities etc. It can be also accessed straight from web browser so that there is no need for installing new software on your desktop.

Below we deliver direct link to our whole repository. It is a open directory for every person who have Google account:

https://code.google.com/p/applications-in-high-end-computing

We are aware that is not perfectly secure way of storing data. But, thanks to SVN we are not afraid of losing permanent data.

d) Microsoft Project 2010

For generating Gantt charts we have been using this, Microsoft software. Generally it offers a lot more but we have used only a small part of its possibilities.

e) Eclipse Indigo Service Release 1

Since we have decided to use Java programming language in our project we have actually choice only between NetBeans and program mentioned above. We have decided to use Eclipse because, in our opinion, is more stable and open for different kinds of plugins. It is also very well documented software which is very important feature of Developers Environment.

8. Last comments

Group project has been great opportunity to cooperate with other students in practical sense, which all of us has very much enjoyed. We have spent together dozens of hours trying to explain ideas within team. It was, especially, valuable lessons of communication skill - which can be priceless experiences during looking for potential jobs.

The only thing which made us sad, was amount of time designed for this assignment. Less than 20 days was not enough for such a complicated project, even if we had nothing else to be done - unfortunately we have been working on two complicated projects simultaneously. On the other hand, this situation forced us to deal with moments of crisis management which could be also useful in the future.

Our final impressions are very positive. Hungarian-Polish cooperation was successful, at least in the sense that after few weeks we are still talking to each other, asking about weather etc.